Journal of Zankoy Sulaimani
Part-A- (Pure and Applied Sciences)
http://jzs.univsul.edu.iq

Competitive Algorithms for Large Scale Positive Definite Linear Systems of Equations Using an Error in Variables Optimization Model

Nezam Mahdavi-Amiri[*1] and Negin Bagherpour[1]

1 Department of Mathematical Sciences, Sharif University of Technology, Tehran, Iran
**Corresponding author's e-mail: nezamm@sharif.edu*

Article info
Original: 29 may 2020
Revised: 21 August 2020
Accepted: 26 September 2020
Published online: 20 December 2020

Key Words: large scale problems, Dolan-More, ScaLAPACK

Abstract

We seek positive definite solutions of large scale overdetermined linear systems of equations with multiple right hand sides, where the coefficient and the right hand side matrices are respectively called data and target matrices. We recently proposed a new approach for solving such problems in small scales considering error in both the measured data and the target matrices and using a novel error formulation. We proposed four algorithms for solving small scale positive definite linear systems of equations with both full rank and rank deficient data matrices. Here, we extend our algorithms to solve large scale problems. The algorithms are implemented in Matlab and Fortran (using the ScaLAPACK library) respectively for small and large scale problems . ScaLAPACK is a parallel library written in Fortran to solve linear algebra problems such as computing matrix decompositions for large scale matrices. Block cyclic data distribution is used in ScaLAPACK to provide proper scalability, reliability, portability, flexibility and ease of use. Comparative numerical experiments with two available methods, the interior point method and a method based on quadratic programming, show that our approach produce a smaller standard deviation of the error entries and a smaller effective rank, as being desirable for control problems. Using the Dolan-More performance profiles we show that our proposed algorithms are more efficient and more accurate in computing proper solutions.

1. Introduction

In different engineering contexts such as estimating the mass inertia matrix in the design of controllers for solid structures and robots ([11], [16], [18]), a symmetric positive definite solution of an overdetermined linear system of equations is desired. Modeling a deformable structure also leads to such a mathematical problem; e.g., [28]. More precisely, a positive definite solution $X \in \mathbb{R}^{n\times n}$ for

$$DX \simeq T, \tag{1}$$

is needed, where $D, T \in \mathbb{R}^{m\times n}$, with $m \geq n$, are given. In some special applications, the data matrix D has a simple structure, which may be considered for efficiently organized computations. Estimation of the covariance matrix and computation of the correlation matrix in finance are two such examples, where the data matrices are respectively block diagonal and the identity matrix [34].

The available least squares formulations for physical problems may be classified as either ordinary or error in variables (EIV) models [27]. Also, least squares problems with multiple right hand sides may arise [3].
Here, we consider EIV models for solving (1). In an EIV model, errors are considered in both D and T [5]. Total least squares formulation is a well-known EIV model, where the goal is to solve the following mathematical problem ([8] and [17]):

$$\min \| [\Delta D, \Delta T] \|$$
$$\text{s.t.} (D + \Delta D)x = t + \Delta T$$
$$X \succ 0. \quad (2)$$

where $\|\cdot\|$ in (2) denotes the vector 2-norm.

We have recently considered a new EIV model for solving a positive definite linear system. Our goal is to compute a symmetric positive definite solution $X \in \mathbb{R}^{n\times n}$ for overdetermined system of equations $DX \simeq T$, where both matrices D and T may contain errors. Hereafter, we refer to this problem as positive definite linear system of equations. No EIV model, even the well-known total least squares formulation, was considered for solving the positive definite linear system of equations in the literature. Several approaches have been proposed for this problem, commonly considering the ordinary least squares formulation and minimizing the error $\| \Delta T \|_F$ over all $n \times n$ symmetric positive definite matrices, where $\|.\|_F$ is the Frobenious norm [3, 12, 26]. Here, we briefly discuss our newly formulated error criterion and the corresponding algorithms. We then extend the algorithms for solving large scale test problems; a summary of the ScaLAPACK library is also provided to show its use in solving large scale problems.

Below, we first review our newly defined error function and the outlined algorithms for both full rank and rank deficient problems. We then describe the ScaLAPACK library and discuss how to parallelize the algorithms specifically for large scale problems. Moreover, we provide numerical results to show the effectiveness of our proposed algorithms in computing positive definite solutions of both small and large scale linear systems. Finally, concluding remarks are presented.

2 Problem Formulation and the Proposed Algorithms

Consider a single equation $ax \simeq b$, where $a, b \in \mathbb{R}^n$ and $x \in \mathbb{R}^+$. As shown in Fig. 1, errors in the ith entry of b and a are respectively equal to $|a_i x - b_i|$ and $|a_i - \frac{b_i}{x}|$ [27].

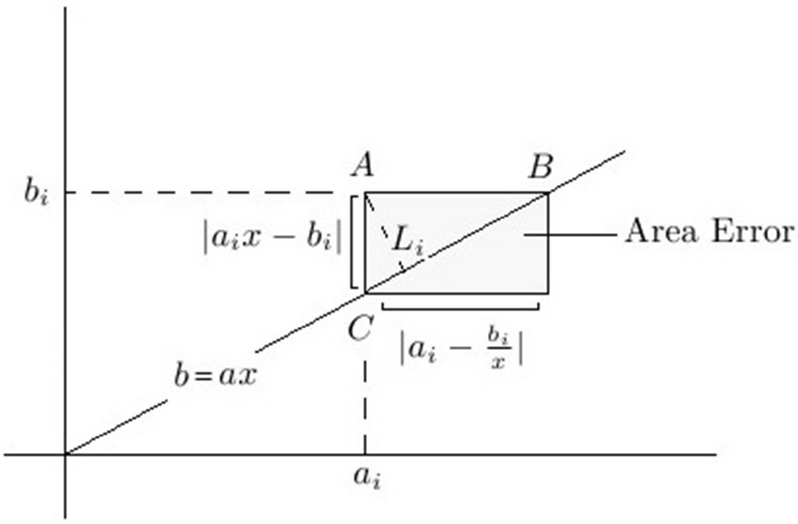

Figure 1: Error formulation for a single equation.

In [27], $\sum_{i=1}^{n} L_i$ was considered as a value to represent errors in both a and b, where L_i is the height of the triangle ABC being equal to $L_i = \frac{|b_i - a_i x|}{\sqrt{1+x^2}}$. Here, to represent the errors in both a and b, we define the area error to be

$$\sum_{i=1}^{n} |b_i - a_i x||a_i - \frac{b_i}{x}|, \quad (3)$$

which is equivalent to

$$\sum_{i=1}^{n}(b_i - a_i x)(a_i - \frac{b_i}{x}),$$

for $x \in \mathbb{R}^+$.

Now, consider the problem of finding a symmetric and positive definite solution to the overdetermined system of linear equations $DX \simeq T$, in which both D and T include errors. The values DX and TX^{-1} as the predicted values for T and D from the model $DX \simeq T$ lead to defining $\Delta T_j = (DX - T)_j$ and $\Delta D_j = (D - TX^{-1})_j$ is the entries of errors in the jth column of T and D, respectively. Extending the area error defined in (3), the value

$$E = \sum_{j=1}^{n}(DX_j - T_j)^T(D_j - (TX^{-1})_j$$

appears to be an appropriate measure of error. We also have

$$E = \sum_{j=1}^{n}\sum_{i=1}^{m}(DX - T)_{ij}(D - TX^{-1})_{ij} = \mathrm{tr}((DX - T)^T(D - TX^{-1})), \tag{4}$$

with $\mathrm{tr}(.)$ standing for trace of a matrix. Therefore, an optimization problem may be formulated as

$$\min_{X \succ 0}\mathrm{tr}((DX - T)^T(D - TX^{-1})), \tag{5}$$

where X is symmetric and by $X \succ 0$, we mean X being positive definite. Problem (5) poses our newly defined EIV model for solving positive definite linear systems of equations. The following note states a property of the formulated error.

Note. A characteristic of the error formulation defined by (4) is that for a symmetric and positive definite matrix X, E is nonnegative and it is equal to zero if and only if $DX = T$.

Now, we proceed to develop an algorithm for solving (5) with the assumption that D has full column rank. With X being symmetric, we have

$$\mathrm{tr}((DX - T)^T(D - TX^{-1})) = \mathrm{tr}(D^TDX + X^{-1}T^TT) - 2\mathrm{tr}(T^TD).$$

So, (5) can be written as

$$\min \mathrm{tr}(AX + X^{-1}B), \tag{6}$$

where $A = D^TD$ and $B = T^TT$ and the symmetric positive definite matrix X is to be computed. To explain our method for solving (6), we state the following theorem from [3].

Throrem 1. *The unique symmetric and positive definite solution for problem (6) is X^* satisfying*

$$X^*AX^* = B. \tag{7}$$

As mentioned in Theorem 1, the KKT conditions lead to the following nonlinear matrix equation

$$XAX = B. \tag{8}$$

Note that (8) is an special case of the continuous time Riccati equation (CARE) [25],

$$A^TXE + E^TXA - (E^TXB + S)R^{-1}(B^TXE + S^T) + Q = 0, \tag{9}$$

with $R = 0$, $E = \frac{A}{2}$ and $Q = -B$. There is a MATLAB routine to solve CARE for arbitrary values of A, E, B, S, R and Q. To use the routine, it is sufficient to type the command

```
X=care(A,B,Q,R,S,E),
```

for the input arguments as in (9). Higham [25] developed an effective method for computing the positive definite solution to this special CARE when A and B are symmetric and positive definite using well-known decompositions. Lancaster and Rodman [31] also discussed solving different types of algebraic Riccati equations. Moreover, they made a perturbation analysis of the matrix equations. Before we proceed further, we outline some well-known results to be used in the development of our algorithms.

Note. (QR decomposition) The QR decomposition [30] of a matrix $A \in \mathbb{R}^{m \times n}$, with $m \geq n$, is a decomposition of the form $A = QR$, where R is an $m \times n$ upper triangular matrix and Q satisfies $QQ^T = Q^TQ = I$. Moreover, if A has full column rank, then R also has full column rank.

Note. (Cholesky decomposition) A Cholesky decomposition [30] of a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$ is a decomposition of the form $A = R^T R$, where R, known as the Cholesky factor of A, is an $n \times n$ nonsingular upper triangular matrix.

Note. (Spectral decomposition) [30] All eigenvalues of a symmetric matrix, $A \in \mathbb{R}^{n\times n}$, are real and there exists an orthonormal matrix with columns representing the corresponding eigenvectors. Thus, there exist an orthonormal matrix U with columns being the eigenvectors of A and a diagonal matrix D containing the eigenvalues such that $A = UDU^T$. Also, if A is positive definite, then all of its eigenvalues are positive, and so we can set $D = S^2$. Thus, spectral decomposition for a symmetric positive definite matrix A is a decomposition of the form $A = US^2U^T$, where $U^TU = UU^T = I$ and S is a diagonal matrix.

2.1 Full rank system

Theorem 2. *[25] Assume $D, T \in \mathbb{R}^{m\times n}$ with $m \geq n$ are known and $rank(D) = rank(T) = n$. Let $D = QR$ be the QR decomposition of D. Let $A = D^TD$ and $B = T^TT$. Define the matrix $\tilde{Q} = RBR^T$ and compute its spectral decomposition, that is, $\tilde{Q} = RBR^T = U\tilde{S}^2U^T$. Then, (6) has a unique solution, given by*

$$X^* = R^{-1}U\tilde{S}U^TR^{-T}.$$

Proof. It is sufficient to show that X^* satisfies $X^*AX^* = B$ [3, 25]. Note that from $D = QR$, we have

$$A = D^TD = R^TQ^TQR = R^TR.$$

Substituting X^*, we have

$$\begin{aligned} X^*AX^* &= R^{-1}U\tilde{S}U^TR^{-T}R^TRR^{-1}U\tilde{S}U^TR^{-T} \\ &= R^{-1}U\tilde{S}^2U^TR^{-T} = R^{-1}RBR^TR^{-T} = B. \end{aligned}$$

Note. To compute R, it is also possible to first compute $A = D^TD$ and then calculate the Cholesky decomposition of A. However, the QR decomposition, due to its stability, is used in **Algorithm 1** below:

Algorithm 1 Solving the EIV model for positive definite linear system using QR decomposition
PROCEDURE PDEIV-QR (*D, T*). 1: Compute the QR decomposition of D: let $D = QR$. 2: Let $\tilde{Q} = RBR^T$, where $B = T^T T$ and compute the spectral decomposition of $\tilde{Q}$ that is, $\tilde{Q} = U\tilde{S}^2U^T$. 3: Set $X = R^{-1}U\tilde{S}U^TR^{-T}$. 4: Set $E = tr((DX^* - T)^T(D - TR^{*-1}))$.

Note that **Algorithm 1** computes the solution of (5) directly.

The following theorem shows that using the spectral decomposition of A, a method similar to the one introduced in [25] is at hand for solving the continuous time Riccati equation.

Theorem 3. *Let $A = D^TD$ and $B = T^TT$ with $D, T \in \mathbb{R}^{m\times n}$, $m \geq n$, and $rank(D) = n$. Let the spectral decomposition of A be $A = US^2U^T$. Define the matrix $\tilde{Q} = SU^TBUS$ and compute its spectral decomposition, $\tilde{Q} = SU^TBUS = \bar{U}\bar{S}^2\bar{U}^T$. Then, the unique minimizer of (6) is*

$$X^* = US^{-1}\bar{U}\bar{S}\bar{U}^TS^{-1}U^T.$$

Proof. Similar to the proof of **Theorem 2**, it is sufficient to show that X^* satisfies $X^*AX^* = B$. Substituting X^*, we have

$$\begin{aligned} X^*AX^* &= US^{-1}\bar{U}\bar{S}\bar{U}^TS^{-1}U^TUS^2U^TUS^{-1}\bar{U}\bar{S}\bar{U}^TS^{-1}U^T \\ &= US^{-1}\bar{U}\bar{S}^2\bar{U}^TS^{-1}U^T = US^{-1}SU^TBUSS^{-1}U^T = B. \end{aligned}$$

We now summarize the steps of our proposed algorithm for computing the symmetric and positive definite solution of a linear system based on spectral decomposition as **Algorithm 2.**

Algorithm 2 Solving the EIV model using spectral decomposition.
Procedure PDEIV-Spec(D,T). 1. Let $A = D^T D$ and compute its spectral decomposition: $A = US^2U^T$. 2. Let $\tilde{Q} = SU^T BUS$, where $B = T^T T$ and compute the spectral decomposition of $\tilde{Q}$, that is, $\tilde{Q} = \tilde{U}\tilde{S}^2\tilde{U}^T$. 3. Set $X^* = US^{-1}\tilde{U}\tilde{S}\tilde{U}^T S^{-1}U^T$. 4. Set $E = \text{tr}\ \ ((DX^* - T)^T(D - TX^{*-1}))$.

In [3], a modification of the algorithms to compute the symmetric and positive definite solution of a rank deficient linear system is discussed in detail. The main parts of computations and the resulting algorithms based on spectral decomposition and complete orthogonal decomposition are discussed next.

2.2 Rank deficient system

Let the spectral decomposition of A be $A = U\begin{pmatrix} S^2 & 0 \\ 0 & 0 \end{pmatrix}U^T$, where $S^2 \in \mathbb{R}^{r\times r}$ is a diagonal matrix having the positive eigenvalues of A as its diagonal entries. Substituting the decomposition in (7), we get

$$X^*U\begin{pmatrix} S^2 & 0 \\ 0 & 0 \end{pmatrix}U^TX^* = B. \tag{10}$$

Since U is orthonormal, (10) can be written as

$$U^TX^*U\begin{pmatrix} S^2 & 0 \\ 0 & 0 \end{pmatrix}U^TX^*U = U^TBU.$$

Then, letting $\tilde{X} = U^TXU$ and $\tilde{B} = U^TBU$, we have

$$\tilde{X}\begin{pmatrix} S^2 & 0 \\ 0 & 0 \end{pmatrix}\tilde{X} = \tilde{B}. \tag{11}$$

Thus, the matrix $X = U\tilde{X}U^T$ is a solution of (5) if and only if $\tilde{X}$ is symmetric positive definite and satisfies (11). Considering the block form $\tilde{X} = \begin{pmatrix} \tilde{X}_{rr} & \tilde{X}_{r,n-r} \\ \tilde{X}_{n-r,r} & \tilde{X}_{n-r,n-r} \end{pmatrix}$, where $\tilde{X}_{rr} \in \mathbb{R}^{r\times r}$, $\tilde{X}_{r,n-r} = \tilde{X}_{n-r,r}^T \in \mathbb{R}^{r\times(n-r)}$ and $\tilde{X}_{n-r,n-r} \in \mathbb{R}^{(n-r)\times(n-r)}$, (11) turns to

$$\tilde{X}_{rr}S^2\tilde{X}_{rr} = \tilde{B}_{rr}, \tag{12a}$$

$$\tilde{X}_{rr}S^2\tilde{X}_{r,n-r} = \tilde{B}_{r,n-r}, \tag{12b}$$

$$\tilde{X}_{n-r,r}S^2\tilde{X}_{r,n-r} = \tilde{B}_{n-r,n-r}. \tag{12c}$$

Let $\overline{D} = S$ and suppose $\overline{T}$ satisfies $\overline{T}^T\overline{T} = \tilde{B}_{rr}$. Consider problem (5) corresponding to the data and target matrices $\overline{D}$ and $\overline{T}$ as follows:

$$\min_{X>0} \text{tr}((\overline{D}X - \overline{T})^T(\overline{D} - \overline{T}X^{-1})). \tag{13}$$

We know from **Theorem 1** that the necessary and sufficient optimality conditions for the unique solution of problem (13) implies (12a). Thus, $\tilde{X}_{rr}$ can be computed using **Algorithm 1** for the input arguments $\overline{D}$ and $\overline{T}$. Substituting the computed $\tilde{X}_{rr}$ in (12b), the following linear system of equations arises:

$$\tilde{X}_{rr}S^2\tilde{X}_{r,n-r} = \tilde{B}_{r,n-r} \tag{14}$$

arises, where $\widetilde{X}_{rr}, S^2 \in \mathbb{R}^{r\times r}$ are known and $\widetilde{X}_{r,n-r} \in \mathbb{R}^{r\times(n-r)}$ is to be computed. Since $\widetilde{X}_{rr}$ is positive definite and S^2 is nonsingular, the coefficient matrix of the linear system (14) is nonsingular and $\widetilde{X}_{r,n-r}$ can be uniquely computed.

It is clear that since $\widetilde{X}$ is symmetric, $\widetilde{X}_{n-r,r}$ is the same as ${\widetilde{X}_{r,n-r}}^T$. Now, we check whether the computed $\widetilde{X}_{n-r,r}$ and $\widetilde{X}_{r,n-r}$ satisfy (12c). Inconsistency of (13) means that there is no symmetric positive definite matrix satisfying (12a)-(12c), and if so, (5) has no solution. Thus, in solving a specific positive definite system with rank deficient data and target matrices using the proposed EIV model, a straightforward method to investigate the existence of solution is to check whether (12c) holds for the given data and target matrices. On the other hand, for numerical results, it is necessary to generate meaningful test problems. Hence, we can investigate the necessary and sufficient conditions for satisfaction of (12c). In this case, $\widetilde{X}_{rr}$, $\widetilde{X}_{r,n-r}$ and its transpose, $\widetilde{X}_{n-r,r}$, are respectively computed from (12a) and (12b). Hence, the only remaining step is to compute $\widetilde{X}_{n-r,n-r}$ so that $\widetilde{X}$ is symmetric and positive definite.

We know that $\widetilde{X}$ is symmetric positive definite if and only if there exists a nonsingular lower triangular matrix $L \in \mathbb{R}^{n\times n}$ so that

$$\widetilde{X} = LL^T, \tag{15}$$

where L is lower triangular and nonsingular. Considering the block forms

$$\widetilde{X} = \begin{pmatrix} \widetilde{X}_{rr} & \widetilde{X}_{r,n-r} \\ \widetilde{X}_{n-r,r} & \widetilde{X}_{n-r,n-r} \end{pmatrix}$$

and

$$L = \begin{pmatrix} L_{rr} & 0 \\ L_{n-r,r} & L_{n-r,n-r} \end{pmatrix},$$

where $L_{n-r,r}$ is an $(n-r)\times r$ matrix and $L_{rr} \in \mathbb{R}^{r\times r}$ and $L_{n-r,n-r} \in \mathbb{R}^{(n-r)\times(n-r)}$ are nonsingular lower triangular matrices, we get

$$\widetilde{X}_{rr} = L_{rr}{L_{rr}}^T, \tag{16a}$$

$$\widetilde{X}_{r,n-r} = L_{rr}{L_{n-r,r}}^T, \tag{16b}$$

$$\widetilde{X}_{n-r,r} = L_{n-r,r}{L_{rr}}^T, \tag{16c}$$

$$\widetilde{X}_{n-r,n-r} = L_{n-r,r}{L_{n-r,r}}^T + L_{n-r,n-r}{L_{n-r,n-r}}^T. \tag{16d}$$

Therefore, to compute a symmetric positive definite $\widetilde{X}$, (16a)–(16d) must be satisfied.

In general, an arbitrary rank deficient linear system may have no positive definite solution. In [3], we discussed how to generate a rank deficient test problem having a positive definite solution. Our proposed algorithm for computing the positive definite solution of rank definient linear systems based on spectral decomposition is summarized as **Algorithm 3** below.

Note. The complete orthogonal decomposition (CoD) may also be used to solve the rank deficient problem [3].

Algorithm 3 Solving the EIV model with rank deficient data and target matrices using spectral decomposition.
Procedure PDEIV-RD-Spec(D,T,δ). 1. Input: δ as the upper bound for absolute error taken to be close to the machine (or user's) zero. 2. Let $A = D^TD$ and compute its spectral decomposition: $$A = U\begin{pmatrix} S^2 & 0 \\ 0 & 0 \end{pmatrix}U^T,$$ and let $\mathrm{rank}(D) = r$.

3. Let $B = T^T T$ and $\tilde{B} = U^T B U$ and define
   $\tilde{B}_{rr} = \tilde{B}(1:r, 1:r)$,
   $\tilde{B}_{r,n-r} = \tilde{B}(1:r, r+1:n)$,
   $\tilde{B}_{n-r,n-r} = \tilde{B}(r+1:n, r+1:n)$
4. Let $\bar{D} = S$, and let $\bar{T}$ be the Cholesky factor of $\tilde{B}_{rr}$, that is, $\tilde{B}_{rr} = \bar{T}^T \bar{T}$.
5. Perform **Algorithm 1** with input parameters $D = \bar{D}$ and $T = \bar{T}$, and let $\tilde{X}_{rr} = X^*$.
6. Solve the linear system (12b) to compute $\tilde{X}_{r,n-r}$ and let $\tilde{X}_{n-r,r} = \tilde{X}_{r,n-r}{}^T$.
7. **If** $\| U_{n-r}{}^T (B U_r (U_r{}^T B U_r)^{-1} U_r{}^T B - B) \| \geq \delta$ **then stop**; (5) has no solution.
8. **else**
9. Let the Cholesky decomposition of $\tilde{X}_{rr}$ be $\tilde{X}_{rr} = \tilde{L}\tilde{L}^T$ and set $L_{rr} = \tilde{L}$.
10. Solve the lower triangular system (16b) to compute $L_{n-r,r}$.
11. Let $L_{n-r,n-r} \in \mathbb{R}^{(n-r)\times(n-r)}$ be an arbitrary nonsingular lower triangular matrix and compute $\tilde{X}_{n-r,n-r}$ using (16d).
12. Let $\tilde{X} = \begin{pmatrix} \tilde{X}_{rr} & \tilde{X}_{r,n-r} \\ \tilde{X}_{n-r,r} & \tilde{X}_{n-r,n-r} \end{pmatrix}$ and $X^* = U\tilde{X}U^T$.
13. Compute $E = \mathrm{tr}\ ((DX^* - T)(D - TX^{*^{-1}}))$.
14. **end if**

3 ScaLAPACK for Parallel Implementation

Here, we briefly review ScaLAPACK and provide some details of implementation. ScaLAPACK is a library of high-performance linear algebra routines for parallel distributed memory machines. ScaLAPACK solves dense and banded linear systems, least squares problems, eigenvalue problems and singular value problems. The key ideas include the use of a block cyclic data distribution for dense matrices and a block data distribution for banded matrices, block-partitioned algorithms to ensure high levels of data reuse and well-designed low-level modular components that simplify the task of parallelizing the high level routines by making the source codes to perform the same as the sequential case [24].

The goals of the ScaLAPACK project are the same as LAPACK, namely, efficiency, scalability, reliability, portability, flexability and ease of use. Many of these goals, particularly portability, are aided by developing and promoting standards, specially for low-level communication and computation routines. By limiting most machine dependencies to three standard libraries called BLAS, or Basic Linear Algebra Subprograms, LAPACK, or Linear Algebra Package, and BLACS, or Basic Linear Algebra Communication Subprograms, these goals are mostly acheived. The library is written in Fortran. The name ScaLAPACK is an acronym for Scalable Linear Algebra PACKage, or Scalable LAPACK. The most recent version of ScaLAPACK is 2.1.0.0, released in November 16, 2019 [24].

To check the efficiency of our proposed algorithms in finding a symmetric positive definite solution of large scale linear systems, ScaLAPACK 2.1.0.0 is used. To implement PDEIV-QR, PDEIV-RD-Spec, PDEIV-RD-Spec and PDEIV-RD-CoD in Fortran using ScaLAPACK, we need to compute QR, SVD and CoD factorizations. More precisely, one QR and one spectral decompositions are needed in **Algorithm 1**, while in each of **Algorithm 2** and **Algorithm 3**, two spectral decompositions are needed and in **Algorithm 4**, one of the spectral decompositions is replaced by a complete orthogonal decomposition. Assuming real input matrices and single percision, the commands for computing the factorizations are shown in Table 1.

In Section 4, comparative numerical results show the effectiveness of our proposed algorithms in producing more accurate solutions more efficiently.

Algorithm 4 Solving the EIV model with rank deficient data and target matrices using CoD.

Procedure PDEIV-RD-COD(D,T,δ).

1. Input: δ as the upper bound for absolute error taken to be close to the machine (or user's) zero
2. Compute the complete orthogonal decomposition of D, that is,
$$\mathrm{D} = \mathrm{U}\begin{pmatrix} \mathrm{R} & 0 \\ 0 & 0 \end{pmatrix}\mathrm{V}^{\mathrm{T}},$$
and let $\mathrm{rank(D)} = \mathrm{r}$.
3. Let $A = D^T D = V_r R^T R V_r^T$, $B = T^T T$ and $\tilde{B} = V^T B V$, where V_r consists of the first r columns of V.
4. Let
$$\widetilde{\mathrm{B}}_{\mathrm{rr}} = \widetilde{\mathrm{B}}(1{:}\,\mathrm{r}, 1{:}\,\mathrm{r}),$$
$$\widetilde{\mathrm{B}}_{\mathrm{r,n-r}} = \widetilde{\mathrm{B}}(1{:}\,\mathrm{r}, \mathrm{r}+1{:}\,\mathrm{n}),$$
$$\widetilde{\mathrm{B}}_{\mathrm{n-r,n-r}} = \widetilde{\mathrm{B}}(\mathrm{r}+1{:}\,\mathrm{n}, \mathrm{r}+1{:}\,\mathrm{n}).$$
5. Let $\bar{D} = R$, assume $\bar{T}$ satisfies $\tilde{B}_{rr} = \bar{T}^T \bar{T}$.
6. Perform **Algorithm 1** with input parameters $D = \bar{D}$ and $T = \bar{T}$, and let $\tilde{X}_{rr} = X^*$.
7. Solve the linear system (12b) to compute $\tilde{X}_{r,n-r}$ and let $\tilde{X}_{n-r,r} = \tilde{X}^T{}_{r,n-r}$.
8. If $\| U_{n-r}{}^T (B U_r (U_r{}^T B U_r)^{-1} U_r{}^T B - B) \| \geq \delta$ **then stop;** (5) has no solution.
9. **else**
10. Let the Cholesky decomposition of $\tilde{X}_{rr}$ be $\tilde{X}_{rr} = \tilde{L}\tilde{L}^T$ and set $L_{rr} = \tilde{L}$.
11. Solve the lower triangular system (16b) to compute $L_{n-r,r}$.
12. Let $L_{n-r,n-r} \in \mathbb{R}^{(n-r)\times(n-r)}$ be an arbitrary nonsingular lower triangular matrix and compute $\tilde{X}_{n-r,n-r}$ using (16b).
13. Let $\tilde{X} = \begin{pmatrix} \tilde{X}_{rr} & \tilde{X}_{r,n-r} \\ \tilde{X}_{n-r,r} & \tilde{X}_{n-r,n-r} \end{pmatrix}$ and $X^* = V\tilde{X}V^T$.
14. Compute $E = \mathrm{tr}\ \ ((DX^* - T)(D - TX^{*-1}))$. .
15. **end if**

Table 1: Factorizations in ScaLAPACK.

Factorization	Command	Algorithm using the command
QR	psgeqr2	step 1 of **Algorithm 1**
Spectral decomposition	dsyevr, dgeev, zheev and zgeev	step 2 of **Algorithm 1**
		step 1 of **Algorithm 2**
		step 2 of **Algorithm 2**
		step 2 of **Algorithm 3**
CoD	psgeqr2 and xtzrzf	step 2 of **Algorithm 4**

4 Numerical Results

Here, some numerical results corresponding to small and large scale test problems are reported. We made use of MATLAB 2019b in a Windows 10 machine with a 3.2 GHz CPU and a 8 GB RAM. We generated random test problems with random data and target matrices. These random matrices were produced using the rand command in MATLAB. We used rand(m,n) to generate an $\mathrm{m} \times \mathrm{n}$ matrix with uniformly distributed random entries in the interval [0,1]. The random test problems were classified into problems with full column rank data matrix and problems with rank deficient data matrix. Moreover, to generate large scale test problems we used $RANDOM - NUMBER$ command in Fortran in which the interval of random entries can be manually tuned.

4.1 Small scale problems

Here, we report the numerical results corresponding to full rank small scale problems with $0 \leq m, n \leq 1000$. For a given matrix size, we generated 50 random test problems and reported the average time, T, and the average error, E, obtained over all problems, in Table 2. To investigate the effect of using the QR and spectral decompositions in our proposed approach, we constructed the Dolan-Moré performance profiles. The Dolan-Moré performance profile [7] compares the performance of different algorithms on solving a set of given problems.

Since a desired feature in estimation of mass inertia matrix is that the standard deviation value of the resulting error matrix in T be as low as possible, by presenting the Dolan-Moré performance profiles, we compare the required times and the standard deviation values of errors in PDEIV-QR and PDEIV-Spec. It can be concluded from the generated performance profiles in Figs. 2 and 3 that the required time by PDEIV-QR is lower than that of PDEIV-Spec.

Also, to confirm the efficiency of our proposed approach in solving positive definite linear systems, we reported the numerical results obtained by the interior point method (IntP), discussed in [28], and the method based on quadratic programming proposed by Hu [12] (HuM). We then constructed the corresponding Dolan-Moré performance profiles in Figs. 4 and 5. These performance profiles show that our proposed approach is able to compute a solution having a smaller value of standard deviation in less computing time. However, considering the ordinary formulation of IntP and HuM, the more complicated optimization problem,

$$\min \| DX - T \|_F^2$$
$$\text{s.t. } X \succ 0,$$

is needed to be solved; hence, a lower computing time in our proposed approach is not surprising.

In Table 2, the average error value, $E = \text{tr}(DX^* - T)^T(D - TX^{*-1})$, and the average required times (in seconds) are reported for PDEIV-QR and PDEIV-Spec. The first two columns of this table give the matrix size, the third to sixth columns give the time and error for PDEIV-QR and the time and error for PDEIV-Spec, respectively.

Table 2: Average time and error values for PDEIV-QR and PDEIV-Spec.

m	n	Time (PDEIV-QR)	E (PDEIV-QR)	Time (PDEIV-Spec)	E (PDEIV-Spec)
100	10	0.0021	1.6191E+002	0.0014	1.6191E+002
100	50	0.0017	7.2274E+002	0.0022	7.2274E+002
100	100	0.0058	1.2388E+003	0.0072	1.2388E+003
1000	100	0.0089	1.6258E+004	0.0104	1.6258E+004
1000	200	0.0434	3.1684E+004	0.0505	3.1684E+004

The reported results in Table 2 show that PDEIV-QR is faster in computing the solution. The Dolan-Moré performance profile for the required times given in Fig. 2 also confirms this result.

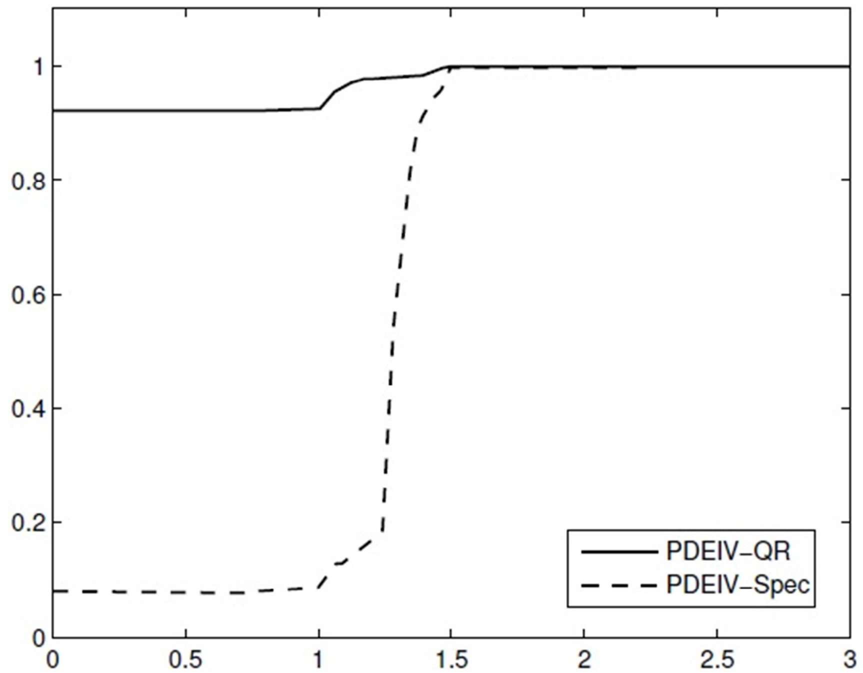

Figure 2: The Dolan-Moré performance profile (comparing the required times by PDEIV-QR and PDEIV-Spec).

However, based on the Dolan-Moré performance profile for the standard deviation values shown in Fig. 3, there is no significant difference between the standard deviation values generated by the two algorithms.

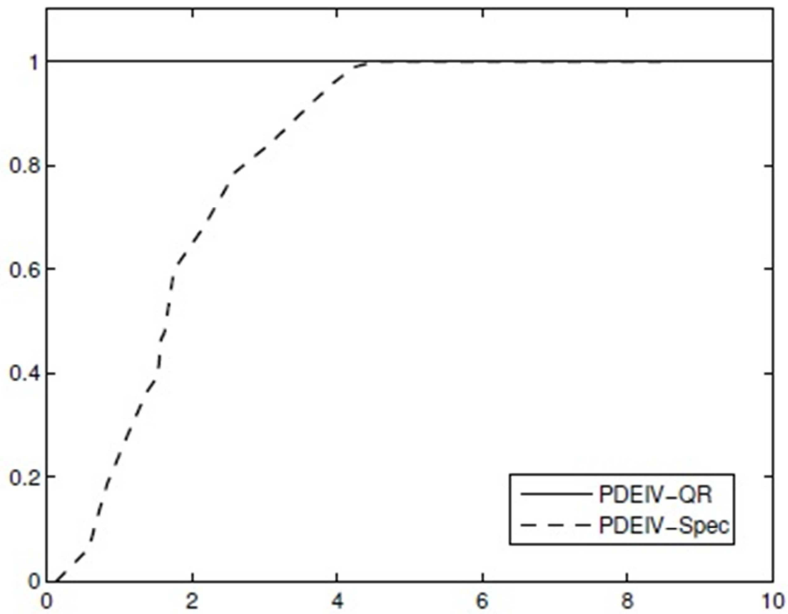

Figure 3: The Dolan-Moré performance profile (comparing the standard deviation values for PDEIV-QR and PDEIV-Spec).

Next, we compare our proposed approach with some other available methods. The Dolan-Moré performance profiles for the times and standard deviation values shown in Figs. 4 and 5 confirm the efficiency of our proposed approach in computing a solution with lower value of standard deviation of error in lower time as compared to IntP and HuM. 1000 random test problems with data and target matrices with dimensions less than 1000×200 were generated to construct the Dolan-Moré performance profiles.

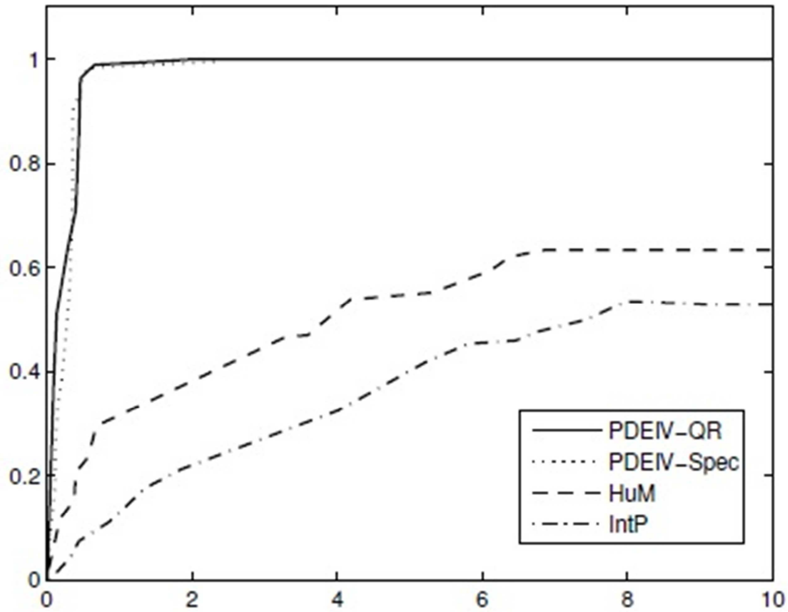

Figure 4: The Dolan-Moré performance profile (comparing the required times by PDEIV-QR, PDEIV-Spec, IntP and HuM).

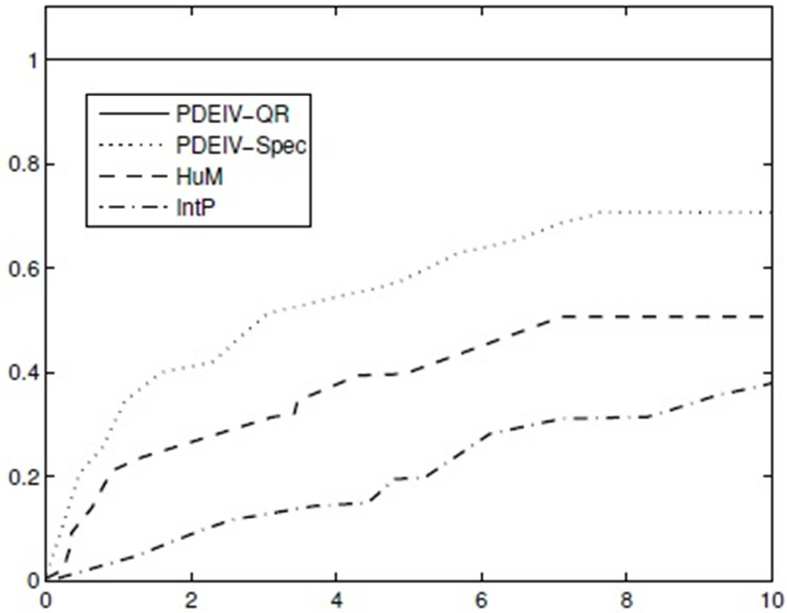

Figure 5: The Dolan-Moré performance profile (comparing the standard deviation values for PDEIV-QR, PDEIV-Spec, IntP and HuM).

Note. There are three important points about the value of error bound, TOL, in the iterative methods IntP and HuM:

(1) Both IntP and HuM as iterative methods need a TOL, the maximum value of possible error, as an input.

(2) Smaller values of TOL result in larger computing times without considerable change in standard deviation values.

(3) For all matrix sizes, except for 1000×200 matrices, the value of TOL was taken to be 10^{-6}. For this case, we set the value of TOL to 0.001 because using a smaller value of TOL, an out of memory error occurred.

4.2 Rank deficient system

In case of rank deficient data and target matrices, we also investigated the effect of using complete orthogonal decomposition and spectral decomposition in the proposed approach. Our proposed approach was also compared with the other available methods based on the Dolan-Moré performance profiles as presented in Figs. 6 and 7. Numerical results also confirm the effectiveness of algorithms 1 through 4 in producing more accurate solutions with lower standard deviation values in lower times. In this section similar results for small scale test problems having rank deficient data and target matrices are reported. In Table 3, we see the numerical results obtained by PDEIV-RD-Spec and PDEIV-RD-COD. In Table 3, the average error values and the required times for PDEIV-RD-Spec and PDEIV-RD-COD are reported. These results show that PDEIV-RD-Spec computes the solution faster, but there is no significant difference in the obtained standard deviation values.

Table 3: Average time and error values for PDEIV-RD-Spec and PDEIV-RD-COD.

m	n	r	Time (Spec)	E (Spec)	Time (COD)	E (COD)
100	10	5	3.6377E-004	1.8733E+002	6.3001E-004	1.8733E+002
100	50	20	1.4125E-003	2.0468E+003	1.6243E-003	2.0468E+003
100	100	50	5.1234E-003	3.9126E+003	5.9146E-003	3.9126E+003
1000	100	50	6.3142E-003	2.0047E+004	1.2843E-002	1.6258E+004
1000	200	100	3.0763E-002	5.8443E+004	4.3702E-002	5.8443E+004

To compare our proposed approach with IntP and HuM, the Dolan-Moré performance profiles corresponding to required times and standard deviations for IntP, HuM, PDEIV-RD-Spec and PDEIV-RD-COD are given in Figs. 6 and 7. To construct these profiles, 1000 random test problems with 1000×100 data and target matrices were generated. The Dolan-Moré profiles confirm that PDEIV-RD-Spec and PDEIV-RD-COD compute solutions with lower values of standard deviation in lower times.

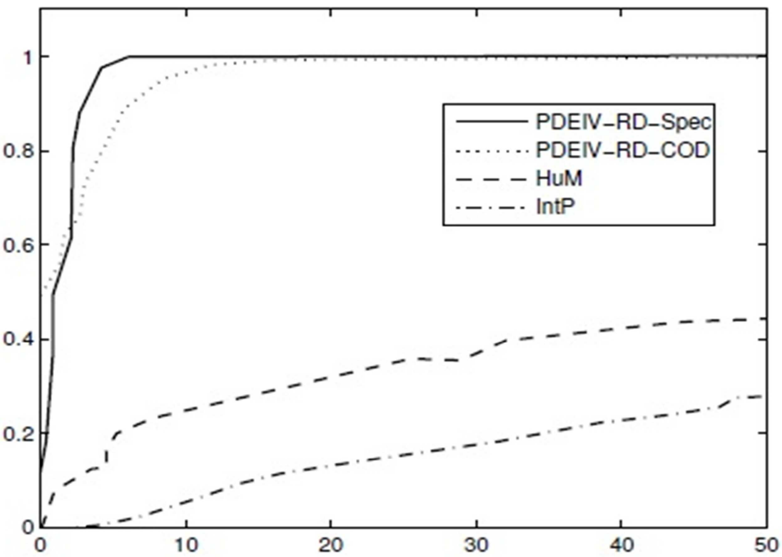

Figure 6: The Dolan-Moré performance profile (comparing the required times by PDEIV-RD-Spec, PDEIV-RD-COD, IntP and HuM).

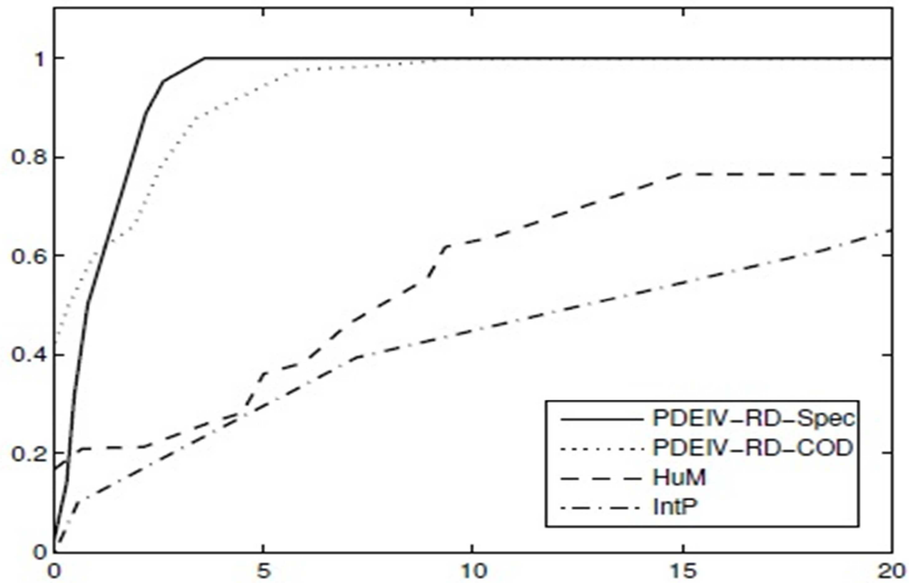

Figure 7: The Dolan-Moré performance profile (comparing the standard deviation values for PDEIV-RD-Spec, PDEIV-RD-COD, IntP and HuM).

The Dolan-Moré performance profiles for effective rank and condition number presented in Figs. 8 and 9 confirm the efficiency of our proposed algorithm in generating solutions with lower values of effective rank and condition number.

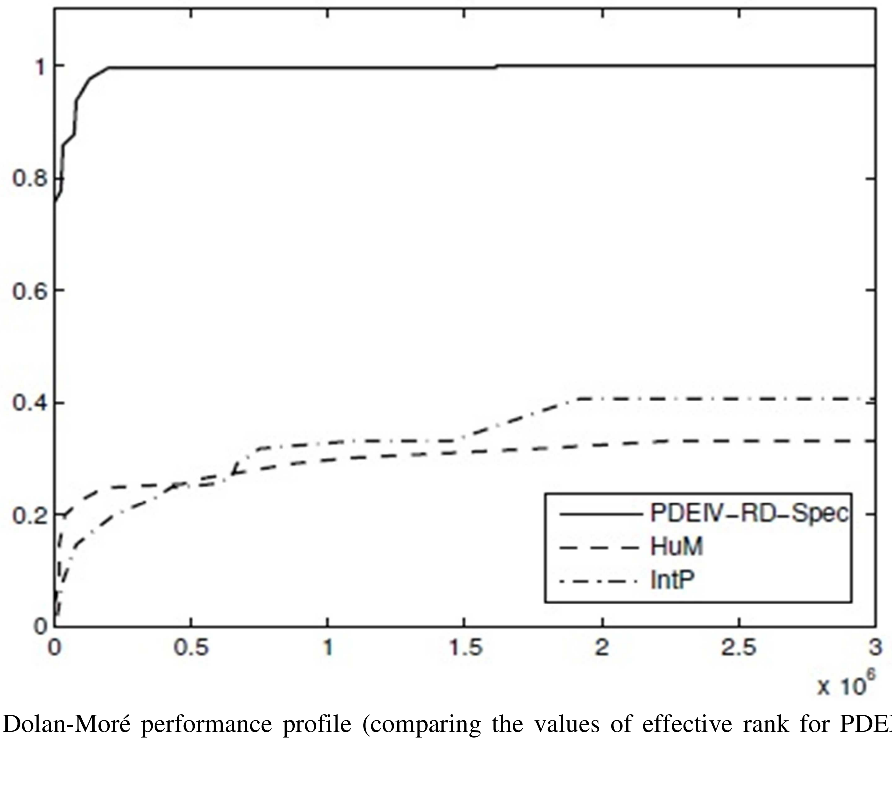

Figure 8: The Dolan-Moré performance profile (comparing the values of effective rank for PDEIV-RD-Spec, IntP and HuM).

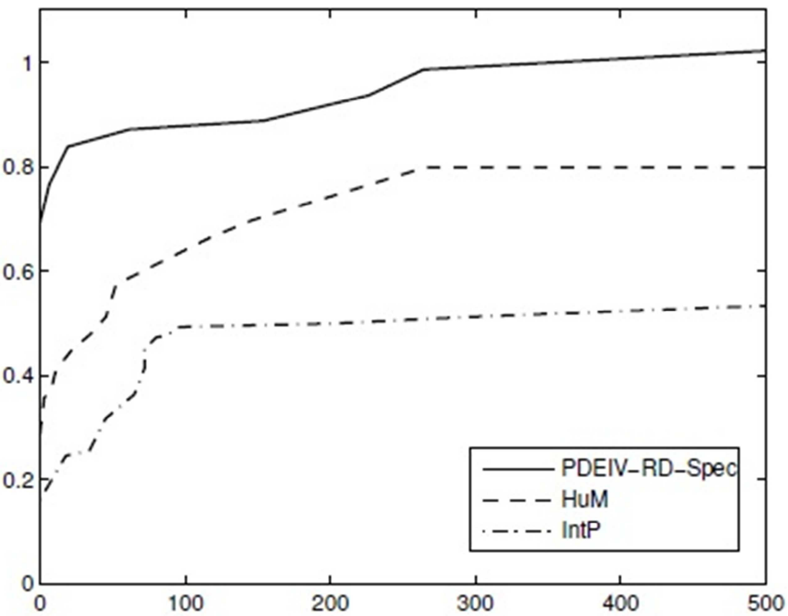

Figure 9: The Dolan-Moré performance profile (comparing the values of condition number for PDEIV-RD-Spec, IntP and HuM).

Next, numerical results are reported to confirm the efficiency of our proposed algorithms in solving large scale positive definite linear systems.

4.3 Large scale problems

Here, numerical results for large scale test problems are presented. A test problem is considered to be large when the data and target matrices are $\mathrm{m} \times \mathrm{n}$ and $5000 \leq \mathrm{m}, \mathrm{n} \leq 20000$. However, according with computational capacity of our computing machine, we only generated test problems with $2000 \leq \mathrm{m}, \mathrm{n} \leq 10000$ as large problems. For a given matrix size and a given rank, we generated 50 test problems and similar to Section 4.1, the average required time and average value of error, E, are reported in Table 3. For large scale tests, algorithms 1-4 were implemented in Fortran and the needed matrix decompositions were computed according to Table 1. In Table 4, the average error value, $\mathrm{E} = \mathrm{tr}(\mathrm{DX}^* - \mathrm{T})^{\mathrm{T}}(\mathrm{D} - \mathrm{TX}^{*-1})$, and the average required times (in seconds) are reported for PDEIV-QR and PDEIV-Spec. The first two columns of this table give the matrix dimensions, the third to sixth columns give the time and error for PDEIV-QR and the time and error for PDEIV-Spec, respectively.

Table 4: Average time and error values for PDEIV-QR and PDEIV-Spec.

m	n	Time (PDEIV-QR)	E (PDEIV-QR)	Time (PDEIV-Spec)	E (PDEIV-Spec)
1000	1000	0.0328	2.7365E+003	0.0564	3.1125E+003
2000	1000	0.0412	6.5981E+003	0.0743	7.2274E+003
2000	2000	0.0583	2.0174E+004	0.0983	3.4253E+004
10000	1000	0.0788	2.7562E+005	0.1642	3.5266E+005
10000	2000	0.1034	4.6583E+005	0.2347	5.6237E+005

The reported results in Table 4 show that PDEIV-QR is faster in computing the solution of large scale systems. To investigate the efficiency of our proposed algorithms for large scale problems, we implemented the interior point algorithm specified for large scale nonlinear optimization problems as outlined by Byrd et al. [4], namely (LS-IntP), the large scale IntP algorithm. As shown in the Dolan-Moré performance profile presented in Figs. 10

and 11, our proposed algorithms are faster and more accurate for solving large scale positive definite linear systems.

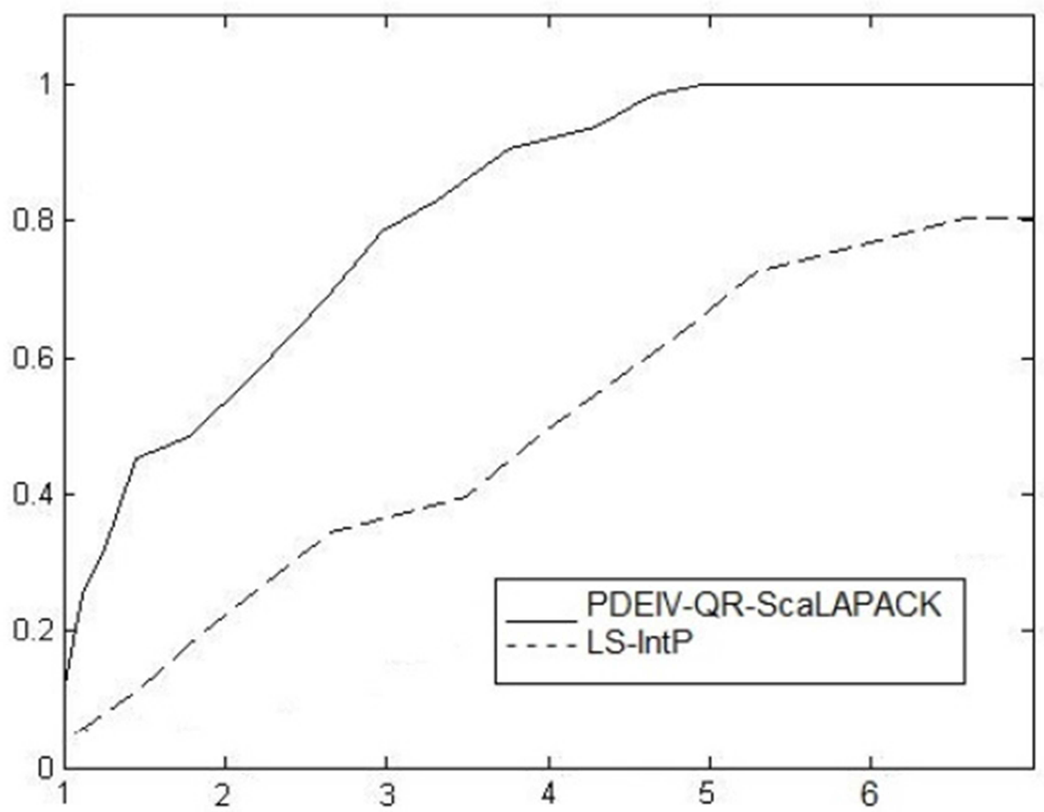

Figure 10: The Dolan-Moré performance profile (comparing the computing times for PDEIV-QR-ScaLAPACK and LS-IntP).

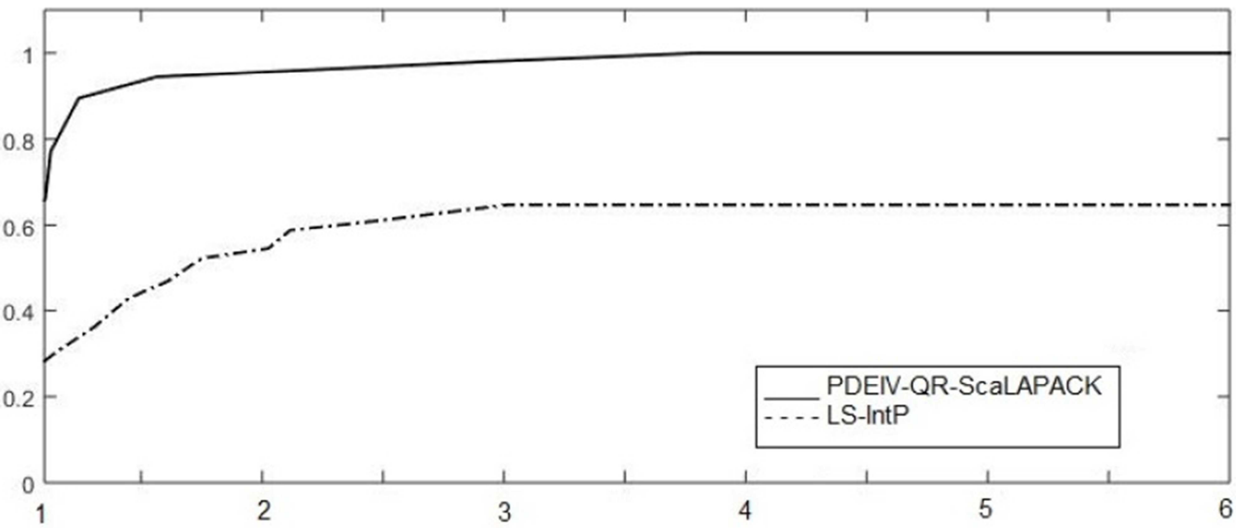

Figure 11: The Dolan-Moré performance profile (comparing the standard deviation values for PDEIV-QR-ScaLAPACK and LS-IntP).

5 Conclusion

Our recently proposed error criterion for solving positive definite linear systems of equations is discussed while the outlines of algorithms for solving such problems using the QR, Spectral and complete orthogonal decompositions are also presented. Considering the fact that positive definite linear systems are not necessarily small in real applications, we applied the algorithms to large scale systems. To this end, an efficient parallel computing platform for matrix decompositions was needed; hence, a brief review of ScaLAPACK and its application in computing matrix factorizations were given. To solve large scale positive definite linear systems, we needed to associate the Fortran implementations with the ScaLAPACK library. Our proposed algorithms for both full rank and rank deficient data and target matrices admit four desirable features including: (1) the proposed modelling is more realistic according to the assumed error in both data and target matrices, (2) the proposed algorithm is a direct, as opposed to iterative, approach, (3) according to the reported numerical results, the obtained average standard deviation value in the error matrix is lower and (4) for large scale test problems, the Fortran implementation is successful in computing proper solutions in lower computing times as compared to other available methods. For large scale test problems, the Fortran (with ScaLAPACK) implementation of our proposed algorithms and a parallel implementation of the interior point algorithm were used to show the efficiency and accuracy of the proposed algorithms.

Acknowledgments

The authors thank the Research Council of Sharif University of Technology for supporting this work.

References

[1] Alizadeh F., Pierre J., Heaberly A., Overton M. L. *"Primal-dual interior point methods for semidefinite programming: convergence rates, stability and numerical result*". SIAM J. Optim. Vol. 8, pp. 746-768. (1998).

[2] Aubry A., Maio A. D., Pallotta L., Farina A. *"Maximum likelihood estimation of a structured covariance matrix with a condition number constraint*". IEEE Trans. On Signal Processing. Vol. 60, No. 6, pp. 3004-3021. (2012).

[3] Bagherpour N., Mahdavi-Amiri N. *"A new error in variables model for solving positive definite linear system using orthogonal matrix decompositions*". Numer. Alg. Vol. 72, No. 1, pp. 211-241. (2016).

[4] Byrd R. H., Mary E., Nocedal J. *"An Interior Point Algorithm for Large-Scale Nonlinear Programming*". SIAM J. Optim. Vol. 9, No. 4, pp. 877-900. (1999).

[5] Cheng C. L., Kukush A., Mastronardi N., Paige C., Van Huffel S. *"Total Least Squares and Errors-in-variables Modeling*". Comput. Stat. Data An. Vol. 52, pp. 1076-1079. (2007).

[6] Deng Y., Boley D. *"On the Optimal Approximation for the Symmetric Procrustes Problems of the Matrix Equation AXB = C*". Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, Chicago. Pp. 159-168. (2007).

[7] Dolan E. D, Moré J. J. *"Benchmarking optimization software with performance profiles*". Mathematical Programming. Vol. 91, pp. 201-213. (2012).

[8] Golub G. H., Van Loan C. F. *"An analysis of the total least squares problem*". SIAM J. Numer. Anal. Vol. 17, pp. 883-893. (1980).

[9] Hayami K., Yin J. F., Ito T. *"GMRES method for least squares problems*". SIAM. J. Matrix Anal. and Appl. Vol. 31, No. 5, pp. 2400-2430. (2010).

[10] Hnětynková I., Plešinger M., Sima D. M., Strakoš Z., Van Huffel S. *"The total least squares problem in $AX \approx B$, A new classification with the relationship to the classical works"*. SIAM J. Matrix Anal. Appl. Vol. 32, No. 3, pp. 748-770. (2011).

[11] Hu H., Olkin I. *"A numerical procedure for finding the positive definite matrix closest to a patterned matrix"*. Statistical and Probability Letters. Vol. 12, pp. 511-515. (1991).

[12] Hu H. *"Positive definite constrained least-squares estimation of matrices"*. Linear Algebra and its Applications. Vol. 229, pp. 167-174. (1995).

[13] Van Huffel S., Vandewalle J. *"Algebraic connections between the least squares and total least squares problems"*. Numer. Math. Vol. 55, pp. 431-449. (1989).

[14] Kang B., Jung S., Park P. *"A new iterative method for solving total least squares problem"*. Proceeding of the 8th Asian Control Conference (ASCC), Kaohsiung, Taiwan, (2011).

[15] Larson H. J. *"Least squares estimation of the components of a symmetric matrix*". Technometrics. Vol. 8, No. 2, pp. 360-362. (1966).

[16] McInroy J., Hamann J. C. *"Design and control of flexure jointed hexapods"*. IEEE Trans. Robotics and Automation. Vol. 16, No. 4, pp. 372-381. (2000).

[17] Paige C. C., Strakoš Z. *"Scaled total least squares fundamentals"*. Numer. Math. Vol. 91, pp. 117-146. (2000).

[18] Poignet P., Gautier M. *"Comparison of Weighted Least Squares and Extended Kalman Filtering Methods for Dynamic Identification of Robots"*. Proceedings of the IEEE Conference on Robotics and Automation, San Francisco, CA, USA, pp. 3622-3627. (2000).

[19] Woodgate K. G. *"Least-squares solution of $F = PG$ over positive semidefinite symmetric P*". Linear Algebra Appl. Vol. 245, pp. 171-190. (1996).

[20] Zhou L., Lin L., Wei Y., Qiao S. *"Perturbation analysis and condition numbers of scaled total least squares problems*". Numer. Algorithms. Vol. 51, pp. 381-399. (2009).

[21] Banerjee S., Roy A. *"Quadratic Forms, Linear Algebra and Matrix Analysis for Statistics*". Chapman & Hall/CRC Texts in Statistical Sciences, pp. 441-442. (2014).

[22] Carlen E. *"Trace Inequalities and Quantum Entropy: An Introductory Course*". Contemporary Mathematics, Vol. 529, AMS, pp. 73-140. (2009).

[23] Gill P. E. , Murray W., Wright M. H. *"Numerical Linear Algebra and Optimization*". Addison Wesley (1991).

[24] Blackford L. S., Choi J., Cleary A., et. al. "*Software, Environments and Tools: ScaLAPACK User's Guide"*, SIAM, Philadelphia. (1997).

[25] Higham N. J. *"Functions of Matrices: Theory and Computation*". SIAM, Philadelphia. (2008).

[26] Horn R. A., Johnson C. R. *"Topics in Matrix Analysis"*. Cambridge University Press. (1991).

[27] Van Huffel S., Vandewalle J. *"The Total Least Squares Problem: Computational Aspects and Analysis*". SIAM, Philadelphia. (1991).

[28] Krislock N. G. *"Numerical Solution of Semidefinite Constrained Least Squares Problems*". M. Sc. Thesis, University of British Colombia. (2003).

[29] Demmel J. W. *"Applied Numerical Linear Algebra*". 3rd edition, SIAM, Philadelphia. (1996).

[30] Golub G. H., Van Loan C. F. *"Matrix Computation"*. 4th edition, JHU Press. (2012).

[31] Lancaster P., Rodman L. *"Algebraic Riccati Equations*". Clarendon Press. (1995).

[32] Magnus J. R., Neudecker H. *"Matrix Differential Calculus with Applications in Statistics and Econometrics*". 2nd edition, John Wiley & Sons. (1999).

[33] Nocedal J., Wright S. J. *"Numerical Optimization*". Springer, New York. (1999).

[34] Higham N. J. *"Computing the nearest correlation matrix (A problem from nance)"*. MIMS EPrint: 2006.70, http://eprints.ma.man.ac.uk/, (2006). Accessed 26 June (2012).

[35] Petersen K. B., Pedersen M. S.: The Matrix Cookbook, http://orion.uwaterloo.ca/hwolkowi/matrixcookbook.pdf, (2008). Accessed 11 January (2013).

[36] Vershynin R.: Introduction to the non-asymptotic analysis of random matrices, http://arxiv.org/pdf/1011.3027v7.pdf, (2011). Accessed 01 February (2013).

[37] American Mathematical Society, Eigenvalues and sums of Hermitian matrices, http://www.ams.org/bookstore/pspdf/gsm132prev.pdf, (2009). Accessed 18 March (2013).